NIST/ITL

the biometric CONSORTIUM

Common Biometric Exchange File Format

Revision History

1st Draft	August 18, 1999
2nd Draft	October 7, 1999

Technical Development Team

Paul Collier, Identicator
Jeff Dunn, NSA
Mark Jerde, Identicator
Larry O’Gorman, Veridicom

Fernando Podio, NIST/ITL
Lawrence Reinert, NSA
Cathy Tilton, SAFLink

Note:
This document is a draft specification of a common biometric exchange file format. The specification is subject to change.

Table of Contents

Foreword

On February 21st 1999, the Information Technology Laboratory of the National Institute of Standards and Technology (NIST) and the US Biometric Consortium sponsored a Workshop to discuss the potential for reaching industry consensus in a common fingerprint template format. As a result of the discussions at the workshop, the participants agreed to develop a “technology-blind” biometric file format that will facilitate handling different biometric types, versions, and biometric data structures in a common way. The proposed effort does not include standardizing the content of these biometric data structures.

The following principles were generally agreed upon during this workshop:

- There is an advantage of a standard file format to facilitate exchange and inter-operability of biometric data.
- This format should include all modalities of biometrics. It should be "technology-blind". That is, it should not bias, encourage, or discourage any particular vendor or biometric technology from another.
- This format should not attempt to translate among different biometric technologies, but merely to identify them and facilitate their co-existence.

The target audience are developers of biometric-based systems and applications and users.

The expected benefits of the specification are: (a) reducing the need for additional software development that will identify different and proprietary biometric data structures supporting multiple biometric types within a system or application and (b) cost savings.

To ensure that the specification is in agreement with other biometric industrial efforts, the work is being coordinated with the BioAPI Consortium and other standardization activities undertaken by the industry, user organizations and the Biometric Consortium. Several BioAPI Consortium members are also participating in the development of this specification.

BioAPI Consortium Liaison:

Larry O’Gorman

Veridicom, Inc.

973-701-8700
log@veridicom

CBEFF Workshops

First Workshop: May 10, 1999

On May 10th, 1999, NIST/ITL and the BC sponsored a Workshop to start development the common biometric exchange file format. The proposed effort does not include standardizing the content of these biometric data structures. During the May 10th Workshop it was agreed by the participants to continue the development of this file format in a second Workshop.

Second Workshop: September 17, 1999

On September 17th, 1999, NIST/ITL and the BC sponsored a second Workshop to discuss the first CBEFF draft specification. Members of the X9.84 working group (Biometric Information and Security Standards) gave a presentation that outlined their requirements. They stated a need for using ASN.1 syntax for describing information specified in the first draft and for encoding the data using standard encoding rules. The reasons for using ASN.1 are as follows:

- ASN.1 is an international standard. If the CBEFF is to become an ISO/international standard it must use ASN.1.
- X9 requires the use of ASN.1 syntax in its specifications. X9 approved standards become National/International standards.
- Users of the technology such as the Banking industry and the Department of Defense have a requirements to follow National/International standards.
- ASN.1 is flexible and is platform independent.
- ASN.1 has been adopted by many information processing standards security such as X.509, SET (Visa & Mastercard), Microsoft Active Directory, etc.

The general consensus of the Workshop was to accept the need for the ASN.1 and to rewrite the specification using ASN.1 syntax.

(Please direct your questions about the ASN.1 syntax used in this draft to Lawrence Reinert, NSA at lareine@alpha.ncsc.mil, (301) 688-0278)

For additional information about the Workshops or the biometric exchange file format, please contact:

Fernando Podio,
Information Technology Laboratory
NIST
Co-Chair Biometric Consortium
Bldg. 225/A248, Gaithersburg, MD 20899
(301) 975-2947
(301) 869-7429 (fax)
fernando.podio@nist.gov

NOTE: Participants in the CBEFF development need to sign a “Participant and/or Contributor’s Agreement”. If you need a copy of the agreement, please contact Fernando Podio.

Common Biometric Exchange File Format

1. Purpose

To establish an industry specification that defines a Common Biometric Exchange File Format and associated metadata that will enable interoperability of biometric-based application programs and systems from different vendors. This biometric file format will facilitate handling different types, versions, and technologies of biometric data in a common way. An application will easily recognize what type of biometric is available in the system, what version number, vendor's name, etc. and will be able to point at the proper biometric data. A common biometric file format may consist of a header that contains information such as file length and biometric types, followed by a block of data (data structure) in unspecified format that can pertain to one of any biometric template types and any other required biometric metadata.

The purpose of the specification is: (a) facilitating interoperability between different biometric technologies; (b) providings forward compatibility for technology improvements; and (c) simplifying the software/hardware integration process.

2. Scope

The specification intends to accommodate any biometric technology. The specification intends to include the definition of format and content for data elements such as: (a) a biometric metadata header that would contain such information as version number, length of data, encrypt, etc. for each biometric type available to the application/system; (b) biometric template data (content not specified); and (c) any other required biometric data or metadata structure. The biometric common format will not attempt to translate data among different biometric technologies, but merely will identify them and facilitate their co-existence. Although it is conceivable that industrial or user groups may agree upon common standard formats within the biometric data structures defined as part of this specification, the proposed effort does not include standardizing the content of these biometric data structures**.**

The current thinking is specifying one type of record:

It includes a Standard Biometric Header (SBH) and Standard Biometric Data Blocks (SBDBs). Each SBDB can include a block header identifying the type of data block and other required information such as the version number, length of data, and the data itself (e.g., data template).

This flexible and expandable file format allows for adapting the implementation to the biometric system or application the developer needs.

An application can use:

- A compliant small file with a minimum description (for small applications or systems that would not require more than a very brief description of the biometric data blocks) or

- A larger file including a larger number of descriptive fields and data structures as required by that application.

3. Conformance

TBD

4. References

- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1994, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

- ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1994, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*

- ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1994, Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.

- ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1994, *Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

- ITU-T Final Draft Amendment 1 to Recommendation X.690 (1999) | ISO/IEC 8825-1:1994, *Information Technology – ASN.1 encoding rules: Specification of Basic Encoding: Final Draft Amendment 1: Relative Object Identifiers*

- ISO/IEC 9594-8:1997 (1997) | ISO/IEC 9594-8:1997 *Information technology — Open Systems Interconnection — The Directory: Authentication Framework.*

- RSA Laboratories. *PKCS #7: Cryptographic Message Syntax Standard.* Version 1.5, November 1993

5. Definitions

AlgorithmIdentifier: An ASN.1 type that identifies an algorithm (by an object identifier) and any associated parameters. This type is defined in [ISO/IEC 8825].

ASN.1: Abstract Syntax Notation One, as defined in [ISO/IEC 8825].

Attribute: An ASN.1 type that identifies an attribute type (by an object identifier) and an associated attribute value. The ASN.1 type **Attribute** is defined in [ISO/IEC 8825].

BSMB – Biometric Specific Memory Block

Certificate: A digitally signed data unit binding a public key to identity information. A specific format for certificates is defined in [*ISO/IEC 9594-8*].

DER: Distinguished Encoding Rules, as defined in [ISO/IEC 8825]

Object Identifier: A sequence of integers that uniquely identifies an associated data object in a global name space administrated by a hierarchy of naming authorities. This is a primitive data type in ASN.1.

Protocol Data Unit (PDU): A sequence of bits in machine-independent format constituting a message in a protocol.

Relative Object Identifier: A proposed ASN.1 type which makes it possible to transmit a Object Identifier value in a more compact form by transmitting only their trailing arcs when the leading arcs can be determined based upon the context of use.

SBH – Standard Biometric Header

6. Abbreviations, Notation, and Acronyms

ANS.1 – Abstract Syntax Notation One.
BSMB – Biometric Specific Memory Block
DER- Distinguished Encoding Rules
OID - Object Identifier
PDU - Protocol Data Unit
[…] – Used to denote a variable length, typically depending upon details of the implementation.
IBIA - International Biometric Industry Association. The IBIA has agreed to be the registration authority for all Object Identifiers and Relative Object Identifiers related to this specification.
SBH – Standard Biometric Header

7. Data Structures

The data structure(s) defined in this document must be able to handle a wide variety of applications. Many current systems, which are using biometric verification, have limited storage media such as barcode, magstripe, and older smart cards. These applications require a data structure with minimum overhead.

Many newer systems are being developed which require a complex structure to incorporate enhanced features such as individual thresholding and reduction of false rejections. The new systems may also require compliance with an internationally accepted data encoding schemes such as Abstract Syntax Notation One (ASN.1).

The data structure within this specification is designed to accommodate these two extremes and intermediate solutions.

In order to achieve these requirements, the specified data structure shall have the following qualities:

- The data must be encoded using a internationally accepted encoding scheme (i.e. ASN.1)
- There must be a bare minimum structure that contains only the information necessary to process biometric data (ie. Minimize the overhead for those applications that don’t need it).
- The Biometric Data type within the structure must be uniquely identified (to identify which algorithm(s) need to perform the matching).
- Allow for current standards bodies to have flexibility to define data within the opaque data structure (e.g., The BioAPI Consortium).
- Allow for future expandability without complicated registration processes.

The proposed data structures which will incorporate the requirements are as follows:

SBH – Standard Biometric Header
BSMB – Biometric Specific Memory Block

The data structure will be presented in two different forms: a generic description of the fields of the data structure and an ASN.1 definition. The generic description is intended to be an illustration of the encoding of those fields, so as to give those not familiar with ASN.1 and DER encoding rules a chance to understand the underlying data conveyed in this structure. The ASN.1 description will discuss the ASN.1 syntax as used by data structure in detail.

7.1 Generic Field Description

This section is included for illustrative purposes only. It is an attempt to explain the fields contained within the structure, without the use of ASN.1 syntax. Refer to section 7.2 for a more accurate and complete description of the fields within this structure.

When encoding a field with Distinguished Encoding Rules (DER) an identifier and length are placed in front the content (the Data). The identifier byte can include information, such as a choice. Note that DER encoding includes length information, therefore the following description will NOT include separate length fields.

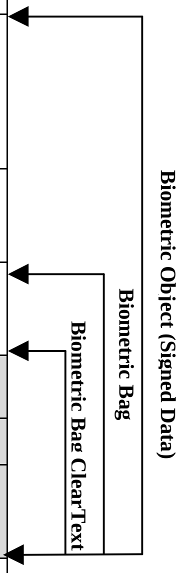
There will be an attempt to illustrate the typical overhead DER brings, because there is a concern about keeping the data structure as small as possible. For this section, the overhead will be defined as the typical length (in bytes) of the DER identifier and length. The overhead is typical, not definite, since DER encoding rules can use different number of bytes to express the length dependant upon the length of data and the method used to encode the length.

7.1.1 Standard Biometric Header (SBH)

The Standard Biometric Header has the fields illustrated in Table 1. The Field name is the ASN.1 label of the given field. The Overhead bytes are the typical number of bytes to represent the DER identifier and length (when it is DER encoded in accordance with section 7.2).

Table 1 - Standard Biometric Header

Field Name	Overhead (bytes)	Notes
SBH (Signature Options)	4	00 = Signed 01 = Unsigned
Header Version	0 or 3	Version of header, should be the version of this specification. Currently set to 00 (the default). This field doesn't exist if the default is used (i.e. overhead=0).
Record Type	0 or 4	Optional, See following definition. This field doesn't exist if it isn't used.
Biometric Bag (Encryption Options)	4	00 = Clear Text 01 = PCKS#7 encrypted data
Format Owner	4	Id of the Group or Vendor which defined the BSMB
Format Type	4	Type as specified by the Owner
Biometric Specific Memory Block (BSMB)	4+[…]	Defined by the Format Owner
Signature Algorithm Identifier	0 or 4+[…]	Only present if the SBH choice is 00
Signature	0 or 4+[…]	Only present if the SBH choice is 00



NOTE:

Not Encrypted
Can be Encrypted

SBH (Signature Options): This field is used to determine if the file is to be signed. A 0 will imply that the data following this field is signed. A 1 will imply the data following this field is unsigned. The Signature field will only be present if the choice specified in this filed is set to 0.

Header Version: This typically 1 byte field allows for future revisions to this specification. Any implementation that wants to be compliant to this version of the specification will set this field to 00. When the default is used, this field is NOT present.

Record Type: This optional field defines the type of biometric this file contains. The currently defined types are as follows:

Field Name	Biometric Type Value
Body Odor	1
DNA	2
Ear Shape	3
Facial Features	4
Finger Image	5
Finger Geometry	6
Hand Geometry	7
Iris Feature	8
Keystroke Dynamics	9
Palm	10
Retina	11
Signature	12
Speech Pattern	13
Vein Pattern	14

This List can be expanded in future revisions of the specification.

Biometric Bag (Encryption Options): This field is used to denote that the Biometric Bag is either encrypted or non-encrypted. A value of 00 indicates that the Biometric Bag is NOT encrypted. The value of 01 denotes that the Biometric Bag is encrypted and formatted in accordance with the EncryptedData description found within PKCS7. If 01 is chosen, the data resulting from the decryption process has the same format as the Clear Text.

Format Owner: This field denotes the Vendor or Group that has defined the format of the Biometric Information and Biometric Data.

Format Type: This field is defined by the Vendor or Group specified in the Vendor or Group name field .

The Biometric Specific Memory Block (BSMB): This contains the biometric data. The Vendor or Group can place a biometric template directly into this field, or it can specify a format for the data with further parameters, information, and data. The BioAPI example in Appendix C is an example of this data can be formatted to contain further information.

Signature Algorithm Identifier: This contains the Algorithm Identifier and any parameters required by the Algorithm. This field exists only if the CBEFF choice is 0.

Signature: This field holds the Signature data. This field exists only if the CBEFF choice is 0.

7.1.2 Biometric Specific Memory Block (BSMB)

This is simply a block of memory that can be specified in any way by the owner of the type as specified in the **Vendor Or Group Name** field of the SBH. Therefore, this can be a proprietary format or one agreed to by a group.

BSMB field format may not need any specification, there is likely to be a format analogous to the header/data format of most data storage structures. In this way, a vendor who "owns" this format can specify information in a header including version information, etc. Furthermore, it is conceivable, or likely, that groups may agree upon common standard formats within this BSMB level. In either case, fields of BSMB can be specified by a vendor or group.

- **BSMB header** - may contain such information as: version number, length of data, encrypt, etc.
- **BSMB template data** - block of memory containing template data.

7.2 ANS.1 Encoding of the Proposed Data Structure

This section is intended to give an overview of the syntax used to describe the proposed data structure using ASN.1. For a complete ASN.1 description refer to Appendix A.

ASN.1 Defines data in terms **Protocol Data Units (PDUs).** Each PDU includes a Identifier, Length, and Content (data). The Value can contain other PDUs (i.e. PDUs can be nested). The Following is a set of definitions for the PDUs used for the Common Biometric File Format. For details on the syntax used to describe these PDUs, refer to [*ISO/IEC 8825-1*].

The following is to be encoded using Distinguished Encoding Rules (DER).

7.2.1 Standard Biometric Header (SBH)

The SBH structure can be signed or unsigned. If the signed option is chosen then the Algorithm Identifier and Signature PDUS will be appended.

```
SBH ::= CHOICE {
                              Signed  [0] Signed{EncodedBiometricObject},
                              UnSigned  [1] BiometricObject   -- RSA PKCS #7 cryptographic type
                    }
```

7.2.1.1 The Biometric Object

The Biometric Object contains the information that can be signed. It holds the version of the PDU, the vendors identifier, processing information and the Biometric data.

```
BiometricObject ::= SEQUENCE {
                                        recordType     RecordType  OPTIONAL,
                                        headerVersion  [0] HeaderVersion DEFAULT hv1,
                                        objectType     ObjectType,
                                        biometricBag   BiometricBag
                                        }
```

7.2.1.1.1 Record Type

The Object types refers to the biometric technology for which this Biometric file is used by. This optional field currently has the following technologies defined:

```
BiometricTypes BIOMETRIC ::= {
                                        { BIOMETRIC id : body-Odor          } |
                                        { BIOMETRIC id : dna                } |
                                        { BIOMETRIC id : ear-Shape          } |
                                        { BIOMETRIC id : facial-Features    } |
                                        { BIOMETRIC id : finger-Image       } |
                                        { BIOMETRIC id : finger-Geometry    } |
                                        { BIOMETRIC id : hand-Geometry      } |
                                        { BIOMETRIC id : iris-Features      } |
                                        { BIOMETRIC id : keystroke-Dynamics } |
                                        { BIOMETRIC id : palm               } |
                                        { BIOMETRIC id : retina             } |
                                        { BIOMETRIC id : signature          } |
                                        { BIOMETRIC id : speach-Pattern     } |
                                        { BIOMETRIC id : vein-Pattern       } |
                                        BiometricTypes-OID,
                                        }
```

Where the BIOMTRIC id has the following (Relative OID) value:

```
body-Odor          RELATIVE-OID ::= { bodyOdor(1) }
dna                RELATIVE-OID ::= { dna(2) }
ear-Shape          RELATIVE-OID ::= { earShape(3) }
facial-Features    RELATIVE-OID ::= { facialFeatures(4) }
finger-Image       RELATIVE-OID ::= { fingerImage(5) }
finger-Geometry    RELATIVE-OID ::= { fingerGeometry(6) }
hand-Geometry      RELATIVE-OID ::= { handGeometry(7) }
```

```
iris-Features       RELATIVE-OID ::= { irisFeatures(8) }
keystroke-Dynamics  RELATIVE-OID ::= { keystrokeDynamics(9) }
palm                RELATIVE-OID ::= { palm(10) }
retina              RELATIVE-OID ::= { retina(11) }
signature           RELATIVE-OID ::= { signature(12) }
speach-Pattern      RELATIVE-OID ::= { speachPattern(13) }
vein-Pattern        RELATIVE-OID ::= { veinPattern(14) }
```

7.2.1.1.2 Header Version

The default and current version is 0. This field represents the version of this specification that this Biometric Object conforms to. Under DER encoding rules, this field is not present if the default is used.

7.2.1.1.3 The Biometric Bag

The Biometric Bag is intended to contain the biometric template information about how to process it. The first piece of processing information determines if the rest of the data has been encrypted. This is accomplished with the following definition:

```
BiometricBag ::= CHOICE {
                             cleartext  [0] Cleartext,
                             encrypted  [1] EncryptedData   -- RSA PKCS #7 cryptographic type
                       }
```

If the data is encrypted then the data is to be decrypted using the data structure defined by PKCS#7 (EncryptedData).   The data recovered from the decryption process has the same format as the Cleartext.

The Cleartext has the following definition:

```
Cleartext ::= SEQUENCE {
                      formatOwner  BIOMETRIC.&name({Owner}),
                      formatType   [0] BIOMETRIC.&Type({Owner}{@formatOwner})  OPTIONAL,
                      bsmb         BSMB  -- Biometric specific memory block
                      }
```

7.2.1.1.3.1 Format Owner

The Format Owner is specifies the Vendor or Group which defined the underlying biometric data. The field must be represented in a compact form, as to not take an unnecessary amount of overhead bytes in order to obtain a unique id. In order to accomplish this, a BIOMETRIC CLASS is defined to contain a name which will be unique:

```
BIOMETRIC ::= CLASS {
                      &name  BIOMETRIC-IDENTIFIER UNIQUE,
                      &Type  OPTIONAL
                      }
 WITH SYNTAX { BIOMETRIC &name [ DATA &Type ] }
```

The name field can be the following BIOMETRIC-IDENTIFIER:

```
BIOMETRIC-IDENTIFIER ::= CHOICE {
                                                  oid  OBJECT IDENTIFIER,       -- complete object identifier
                                                  id   RELATIVE-OID             -- object identifier fragment
                                          }
```

Where the data can be a full Object Identifier or a more compact Relative OID. The Relative OID unique number assigned by the organization maintaining the list (currently assumed to be the IBIA.). The "starting node" of the Relative OID is defined as follows:

```
biometric-id OBJECT IDENTIFIER ::= { x9-84 biometricFormatOwner(1) }
```

This starting node can be used to maintain a "list" of Relative IDs which will be managed by the Controlling Authority (i.e. the IBIA). The initial list will include No Format, The BioAPI, and X9.84.

7.2.1.1.3.2 Format Type

The Format Type specifies the type of biometric data which is to be found within the Biometric Specific Memory Block (BSMB). This field must be represented in a compact form, as to not take an unnecessary amount of overhead bytes in order to obtain a unique id. In order to accomplish this, the RELATIVE-OID will be used in the same manner as the Format Owner as specified in section 7.2.1.1.3.1 with the exception that the "starting node" of the Relative OID is defined as follows:

```
biometric-id OBJECT IDENTIFIER ::= { x9-84 biometricFormatType(2) }
```

This starting node can be used to maintain a second "list" of Relative IDs which will be managed by the Controlling Authority (i.e. the IBIA).

7.2.2 Biometric Specific Memory Block (BSMB)

This is simply a block of memory. It is defined simply as follows:

```
BSMB::=OCTET STRING (Size(1..MAX).
```

This data can be further formatted. The Format Owner and Format Type specify the details of the data within this block. See appendix C and apendix D for examples of BSMB data formats.

7.3 The BioAPI Specification Version .1 BSMB Format

The BioAPI Steering Committee has released the data structure specified in the BioAPI Specification Version 0.1 as a contribution to the Common Biometric Exchange File Format effort. BioAPI Consortium members can participate in the development of the Common Biometric Exchange File Format. It is anticipated that the BioAPI will create a BSMB compliant with this specification. Refer to Appendix C of this specification for further detail.

7.4 The X9.84 Specification Version .1 BSMB header Extension

X9.84 is the standards committee chartered to define Biometric Information Management and Security. The BSMB defined in Appendix D is meant to illustrate a possible definition for the Biometric data that will help meet their security requirements. Refer the X9.84 standards, when they are available.

8. Summary

A framework is shown here for a common biometric exchange file format. The file header defined in this document can be used by a wide variety of applications and can be expanded as needed within the conformance requirements of this specification.

APPENDIX A: The CBEFF ASN.1 Module

A. THE CBEFF ASN.1 Module

```
X9-84-Biometrics DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS All;

IMPORTS EncryptedData FROM PKCS-7;

StandardBiometricHeader ::= SBH

SBH ::= CHOICE {
   signed    [0] SignedBiometricObject,
   unsigned  [1] BiometricObject
}

SignedBiometricObject ::= SIGNED { EncodedBiometricObject }

EncodedBiometricObject ::= BIOMETRIC.&Type( BiometricObject )

BiometricObject ::= SEQUENCE {
   recordType     RecordType OPTIONAL,
   headerVersion  [0] HeaderVersion DEFAULT hv1,
   objectType     ObjectType,
   biometricBag   BiometricBag
}

RecordType ::= INTEGER (0..MAX)

HeaderVersion ::= INTEGER { hv1(0) } (0..MAX)

ObjectType ::= BIOMETRIC.&name({BiometricTypes})

BiometricTypes BIOMETRIC ::= {
   { BIOMETRIC id : body-Odor          } |
   { BIOMETRIC id : dna                } |
   { BIOMETRIC id : ear-Shape          } |
   { BIOMETRIC id : facial-Features    } |
   { BIOMETRIC id : finger-Image       } |
   { BIOMETRIC id : finger-Geometry    } |
   { BIOMETRIC id : hand-Geometry      } |
   { BIOMETRIC id : iris-Features      } |
   { BIOMETRIC id : keystroke-Dynamics } |
   { BIOMETRIC id : palm               } |
   { BIOMETRIC id : retina             } |
   { BIOMETRIC id : signature          } |
   { BIOMETRIC id : speach-Pattern     } |
   { BIOMETRIC id : vein-Pattern       },

   ...  -- expect additional biometric types --
}

-- The BiometricBag contains a vendor
-- header and associated biometric data. The vendor header consists
-- of a "vendorType" which identifies a specific vendor product and
-- an optional "vendorInfo" which may be used to carry associated
-- product information defined as any ASN.1 type. The "biometricData"
-- is an opaque hexadecimal string that may contain a proprietary data
-- format or the result of encoding any ASN.1 type. The BSMB may be
-- protected by encryption when data confidentiality is required for
-- the biometric data.

BiometricBag ::= CHOICE {
   cleartext  [0] Cleartext,
   encrypted  [1] EncryptedData   -- RSA PKCS #7 cryptographic type
}

Cleartext ::= SEQUENCE {
   formatOwner  BIOMETRIC.&name({Owner}),
   formatType   [0] BIOMETRIC.&Type({Owner}{@formatOwner})  OPTIONAL,
   bsmb         BSMB  -- Biometric specific memory block
}

-- A vendor specific identifier may be a relative object identifier
-- defined under the base "BiometricTypes" (a subtype), or a vendor
-- or group object identifier.

Owner BIOMETRIC ::= {
   { BIOMETRIC id : { 0 } DATA OCTET STRING } |            -- No Format
   { BIOMETRIC id : { 1 } BIOAPI } |                       -- Refer to Appendix C
   { BIOMETRIC id : { 2 } X9.84 },                         -- Refer to Appendix D
   ...  -- expect additional vendor specific types --
}

BSMB ::= OCTET STRING(SIZE(1..MAX))

-- When the "encrypted" choice alternative of "BiometricBag" is used,
-- the "encryptedContent" component of type "EncryptedData" contains
-- the result of encrypting the DER (encoded) representation of a
-- value of type "Cleartext", using an ANSI approved algorithm.

EncodedCleartext ::= BIOMETRIC.&Type( Cleartext )

-- biometric information object identifier

x9standard OBJECT IDENTIFIER ::= {
   iso identified-organization tc68(133) country(16) x9(840) 9 }

x9-84 OBJECT IDENTIFIER ::= { x9standard x9-84(84) }

-- biometric information object class

BIOMETRIC ::= CLASS {
   &name  BIOMETRIC-IDENTIFIER UNIQUE,
   &Type  OPTIONAL
}
  WITH SYNTAX { BIOMETRIC &name [ DATA &Type ] }

BIOMETRIC-IDENTIFIER ::= CHOICE {
   oid  OBJECT IDENTIFIER,        -- complete object identifier
   id   RELATIVE-OID              -- object identifier fragment
}

-- biometric technology

body-Odor           RELATIVE-OID ::= { bodyOdor(1) }
dna                 RELATIVE-OID ::= { dna(2) }
ear-Shape           RELATIVE-OID ::= { earShape(3) }
facial-Features     RELATIVE-OID ::= { facialFeatures(4) }
finger-Image        RELATIVE-OID ::= { fingerImage(5) }
finger-Geometry     RELATIVE-OID ::= { fingerGeometry(6) }
hand-Geometry       RELATIVE-OID ::= { handGeometry(7) }
iris-Features       RELATIVE-OID ::= { irisFeatures(8) }
keystroke-Dynamics  RELATIVE-OID ::= { keystrokeDynamics(9) }
palm                RELATIVE-OID ::= { palm(10) }
retina              RELATIVE-OID ::= { retina(11) }
signature           RELATIVE-OID ::= { signature(12) }
speach-Pattern      RELATIVE-OID ::= { speachPattern(13) }
vein-Pattern        RELATIVE-OID ::= { veinPattern(14) }

-- The "biometric-id" object identifier is the base identifier
-- or root of a tree of biometric technology types. Where needed
-- the "BiometricTypes" information object set can carry either
-- relative or complete object identifiers.

biometric-id OBJECT IDENTIFIER ::= { x9-84 biometricFormatOwner(1) }      -- for owner Relative IDs
biometric-id OBJECT IDENTIFIER ::= { x9-84 biometricFormatType(2) }       -- for format types

-- Authentication Information (AI)

BiometricInfo ::= SEQUENCE SIZE(1..MAX) OF BioInformation

BioInformation ::= SEQUENCE {
   processingInfo  [0] ProcessingInfo  OPTIONAL,
   matchingInfo    [1] MatchingInfo
}

-- biometric processing algorithms

-- The biometric processing information type specifies the
-- processing algorithm used to create a given biometric
-- template and any associated process specific parameters.

ProcessingInfo ::= SEQUENCE SIZE(1..MAX) OF ProcessingInformation

ProcessingInformation ::= SEQUENCE {
   id    BIOMETRIC.&name({ProcessingAIDs}),
   parms BIOMETRIC.&Type({ProcessingAIDs}{@id}) OPTIONAL
}

ProcessingAIDs BIOMETRIC ::= {
   { BIOMETRIC oid : processing },
   ... -- expect others --
}
```

```
-- The "processing" object identifier is the base identifier
-- or root of a tree of biometric processing algorithms. It
-- may also identify a default algorithm in contexts where
-- interoperability is not required, or when it is necessary
-- to identify biometric processing algorithms in general.

processing OBJECT IDENTIFIER ::= { x9-84 algorithms(3) }

-- biometric matching methods

MatchingInfo ::= SEQUENCE SIZE(1..MAX) OF MatchingInformation

MatchingInformation ::= SEQUENCE {
  id     BIOMETRIC.&name({MatchingAIDs}),
  parms  BIOMETRIC.&Type({MatchingAIDs}{@id}) OPTIONAL
}

MatchingAIDs BIOMETRIC ::= {
  { BIOMETRIC oid : matching-method },
  ... -- expect others --
}

-- The "matching-method" object identifier is the base
-- identifier or root of a tree of biometric matching functions
-- (algorithms). It may also identify a default algorithm in
-- contexts where interoperability is not required, or when it
-- is necessary to identify matching functions (algorithms) in
-- general.

matching-method OBJECT IDENTIFIER ::= { x9-84 biometricMatch(4) }

-- useful parameterized types --

SIGNED { ToBeSigned } ::= SEQUENCE {
  toBeSigned  ToBeSigned,
  algorithm   AlgorithmIdentifier{{SignatureAlgorithms}},
  signature   BIT STRING
}

AlgorithmIdentifier { ALGORITHM-ID:IOSet } ::= SEQUENCE {
  algorithm   ALGORITHM-ID.&id({IOSet}),
  parameters  ALGORITHM-ID.&Type({IOSet}{@algorithm}) OPTIONAL
}

SignatureAlgorithms ALGORITHM-ID ::= {
  ...  -- any ANSI approved signature algorithm --
}

-- useful information object classes --

ALGORITHM-ID ::= CLASS {
  &id    OBJECT IDENTIFIER  UNIQUE,
  &Type  OPTIONAL
}
 WITH SYNTAX { OID &id [PARMS &Type] }
```

```
END  -- X9-84-Biometrics --


PKCS-7 DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- NOTE: This is a stub used just to resolve references for
--      my syntax checking tool. This stub mimics types
--      defined and used in various versions of PKCS #7,
--      but does not constitute a complete module and
--      should not be included in the X9.84 standard.

EncryptedData ::= SEQUENCE {
  version               INTEGER,
  encryptedContentInfo  EncryptedContentInfo
}

EncryptedContentInfo ::= SEQUENCE {
  contentType                 ContentType,
  contentEncryptionAlgorithm  CEAlgorithmIdentifier,
  encryptedContent            [0] IMPLICIT EncryptedContent OPTIONAL
}

ContentType ::= OBJECT IDENTIFIER

CEAlgorithmIdentifier ::= SEQUENCE {
  algorithm   OBJECT IDENTIFIER,
  parameters  NULL
}

EncryptedContent ::= OCTET STRING

END  -- PKCS-7 --
```

APPENDIX B: An ASN.1 Example

B. Hexadecimal representation of a CBEFF ASN.1 encoded data object

TBD.

APPENDIX C: The BIOAPI Structure; A Biometric Specific Memory Block example

C. Introduction - Data structure proposed for the Common Biometric Exchange File

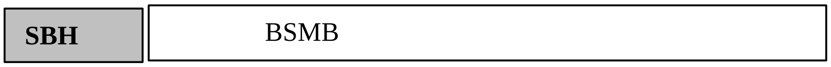
The following is for illustrative purposes only, refer to the BIOAPI documentation when they are available.

This data structure is one that has been used in many fields involving data exchange; a single, technology-neutral header followed by a technology-specific data block. (It has been included in this proposal with the understanding that this is only an example of how the file format to be developed under this effort might look like. It was included to encourage further discussions on the content of the required format.)

The format would look like,

SBH - Standard Biometric Header
BSMB - Biometric Specific Memory Block

Standard Biometric Header (SBH)



- **File Identifier Value ("magic number") -** first field of file identifies this as a biometric exchange file format (BEFF) type; a file identifier value is usually a long integer value or a short string of character bytes in length.

- **Length -** indicates length of data portion of file, BSMB, in bytes.

- **Type -** indicates BSMB type, e.g. "Company X", "Group Y", "Standard Z".

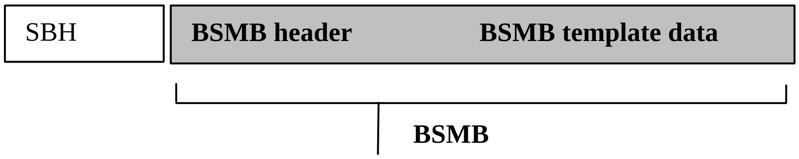
- **Time Stamp -** date and time (it is conceivable that a time stamp also appear in the BSMB data portion, which can be encrypted, so as to safeguard tampering of this information).

- **Encrypt Flag/Type -** indicates if the BSMB data portion is encrypted, and if so, by what method (this may or may not be needed in addition to the ability to encrypt within a portion of the BSMB, see below).

Biometric Specific Memory Block (BSMB)

This is simply a block of memory that can be specified in any way by the owner of the type as specified in the *type* field of the SBH. Therefore, this can be a proprietary format or one agreed to by a group.

BSMB field format may not need any specification, there is likely to be a format analogous to the header/data format of most data storage structures. In this way, a vendor who "owns" this format can specify information in a header including version information, etc. Furthermore, it is conceivable, or likely, that groups may agree upon common standard formats within this BSMB level. In either case, fields of BSMB can be specified by a vendor or group.



- **BSMB header** - contains a length of data.

- **BSMB template data** - block of memory containing template data.

The example consist of a header that contains the data length and biometric type, followed by a block of data in unspecified format that can pertain to one or more of any biometric template types.

C.1 Data Structure Defined in the BioAPI Specification Version 0.1

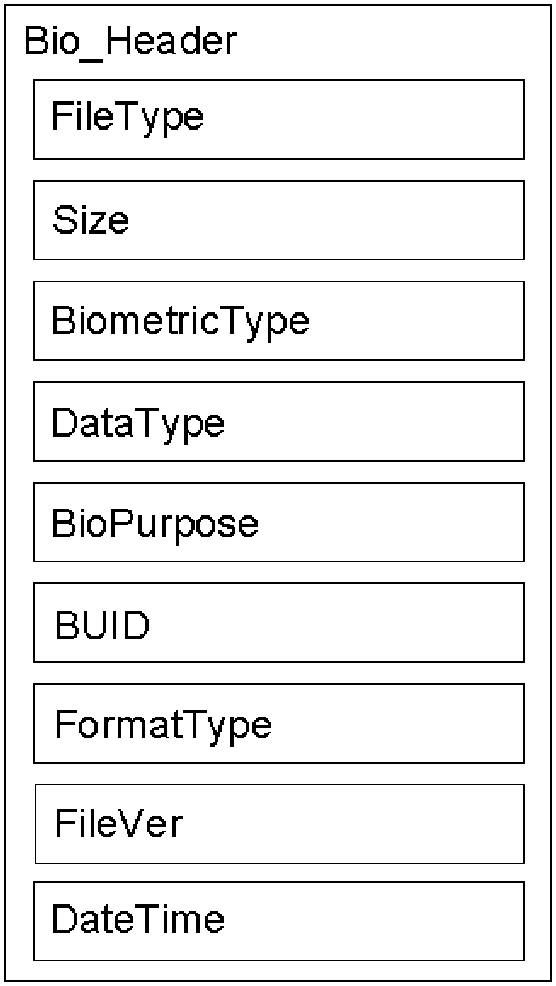
Note:
The BioAPI Steering Committee has released the data structure specified in BioAPI Specification Version 0.1 as a contribution to the Common Biometric Exchange File Format effort. BioAPI Consortium members can participate in the development of the Common Biometric Exchange File Format.

The data structures herein defined have been designed to be as flexible as possible, allowing the biometric vendor to store whatever information is needed, without unnecessary constraints. For example, the biometric data structures may contain a single biometric sample or may contain multiple samples. In order to support a wide range of process flow possibilities and biometric templates (models), these structures can be used to store any combination of data necessary to facilitate subsequent matching. It is the responsibility of the biometric technology provider (BSP) to fill this data structure with the data needed and in the format needed, and to be able to extract this data when it is needed.

C.1.1 Biometric Record Header

This data structure standardizes the header information preceding biometric data records to minimally and uniquely identify the content as well as to distinguish it from other, non-biometric data records.

Table A.1. Biometric record header



Member Description

FileType Unique, assigned value indicating that this as a Biometric Record. Constant string value = BIO.

Size Unsigned long value indicating the size, in bytes, of the opaque biometric data block following the Bio_Header

BiometricType Double word indicating biometric type (fingerprint, face, voice, iris, signature, hand, etc.)

DataType Double word indicating data type as raw, intermediate, or processed.

BioPurpose Double word indicating purpose of biometric data as identification or verification.

BUID The BUID of the BSP which generated the data; stored in as a GUID structure (4 32-bit words).

FormatType Double word indicating unique, specific format of biometric data; stored in OID format.

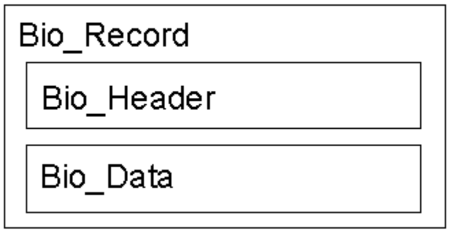
FormatRev Revision level of FormatType specification (double word).

DateTime The date and time at which the data structure was created or last updated. Format TBD.

C.1.2 Biometric Data

This data structure is vendor dependent; however, it is recommended that the vendor include header information within the data structure to facilitate its identification. BSPs are strongly encouraged to document standard data types and formats within the raw biometric data or BIR structures, to facilitate and allow applications to make use of this data. Specifically, if raw biometric data is stored in BMP, TIFF, JPEG, GIF, WAV, AU or other standard formats, documentation should allow for the extraction and display of individual records.

Table A.2. Biometric data



Member Description

Bio_Header Standard biometric record header

Bio_Data An opaque block of data containing raw or processed biometric data samples or templates (biometric identifier records – BIRs). The opaque data should begin with a vendor specific header further describing the contents. This subheader could be defined by another data format standard. Bio_Data can contain a combination of raw and processed data or may contain data captured from more than one biometric capture device.

APPENDIX D: The X9.84 Structure; A Second Biometric Specific Memory Block example

D. The X.984 Structure

D.1 Introduction

The following is for illustrative purposes only, refer to the X9.84 standards, when they are available.

X9.84 is the standards committee chartered to define Biometric Information Management and Security. The BSMB defined in Appendix D is meant to illustrate a possible definition for the Biometric data that will help meet their security and standards related requirements.

X9.84 has the requirement to use ASN.1 syntax to describe all information. Some processing information may need to accompany the Biometric data in order to process the data correctly. The following data structure is to be used as the BSMB when encoding or decoding a CBEFF object which compatible with the X984 specification:

```
X984Biometric::=SEQUENCE{
                                    BiometricInfo BiomtricInfo [OPTIONAL],
                                    BiometricData     BiometricData
                                }
```

Where BiometricData::=OCTET STRING (Size(1..MAX).

The BiometricData is the Raw Biometric Template and the BiometricInfo is defined in the following section.

D.1.1 Optional Biometric Information

The need to identify this process is negligible from the Biometric Objects point of view, unless the process creating the livescan sample to compare against the certificate requires some customizing in regard to the individual who is being sampled. The following definition contains the information for such processing.

```
BiometricInfo ::= SEQUENCE SIZE(1..MAX) OF BioInformation

BioInformation ::= SEQUENCE {
                                        processingInfo  [0] ProcessingInfo  OPTIONAL,
                                        matchingInfo    [1] MatchingInfo
                                    }

-- biometric processing algorithms

-- The biometric processing information type specifies the
-- processing algorithm used to create a given biometric
-- template and any associated process specific parameters.
```

D.1.2 Processing Information

Biometric processing is the function which takes a biometric sample (typically a video image or an audio sample), extracts information from the sample (such as a location of the minutia in the fingerprint), and creates a output file (typically called a biometric template). The processing information field would be used to provide processing algorithm specific information which may be used to personalize the process for the individual. The algorithm used to create the biometric template is specified by the processingAlgorithmID. ProcessingAIDs is used to provide process specific parameters.

```
ProcessingInfo ::= SEQUENCE SIZE(1..MAX) OF ProcessingInformation

ProcessingInformation ::= SEQUENCE OF {
                                          id    BIOMETRIC.&name({ProcessingAIDs}),
                                          parms BIOMETRIC.&Type({ProcessingAIDs}{@id}) OPTIONAL
                                        }

ProcessingAIDs BIOMETRIC ::= {
                                { BIOMETRIC oid : processing },
                                  ... -- expect others --
                                }

-- The "processing" object identifier is the base identifier
-- or root of a tree of biometric processing algorithms. It
-- may also identify a default algorithm in contexts where
-- interoperability is not required, or when it is necessary
-- to identify biometric processing algorithms in general.

processing OBJECT IDENTIFIER ::= { x9-84 algorithms(2) }
```

Examples of processing parameters may be:

> Minimal Acceptable Quality: A minimum quality that the sample must have to be accepted for further processing (useful if the particular biometric can obtain preliminary quality ratings on a sample). This may relieve the need for users with poor biometric characteristics (such as a scarred finger) to reenter a biometric sample several times for verification.
>
> Number of Samples: The number of samples that should be taken of the user which meet the MinimumAcceptableQuality threshold. This will also help users with poor biometric characteristics to avoid reentering a biometric sample several times.

D.1.3 Matching Information

Biometric matching is the function (algorithm) which takes two biometric templates and compares them for similarities. The output of the matching function is typically a matching score representing the amount of similarity found between the two templates.

The biometric templates are generally designed to work with a specific biometric matching algorithm. The application can reference the ID of the MatchingInfo in this field to determine compatibility.

```
MatchingInfo ::= SEQUENCE SIZE(1..MAX) OF MatchingInformation

MatchingInformation ::= SEQUENCE OF {
                                                   id     BIOMETRIC.&name({MatchingAIDs}),
                                                   parms  BIOMETRIC.&Type({MatchingAIDs}{@id}) OPTIONAL
                                                 }

MatchingAIDs BIOMETRIC ::= {
                                         { BIOMETRIC oid : matching-method },
                                         ... -- expect others --
                                       }

-- The "matching-method" object identifier is the base
-- identifier or root of a tree of biometric matching functions
-- (algorithms). It may also identify a default algorithm in
-- contexts where interoperability is not required, or when it
-- is necessary to identify matching functions (algorithms) in
-- general.

matching-method OBJECT IDENTIFIER ::= { x9-84 biometricMatch(3) }
```

Examples of Matching parameters may be:

> Matching Algorithm: A Relative OID which specifies the Algorothm to be used for matching the processed image against a template.
>
> Individual threshold: The minimum matching score required for the user. This may be a useful parameter for those users in which the particular biometric technology has a problem with verification.

D.2 Registering Biometric Processes

If this is the case, then the individual process creating the template needs to be registered by a recognized standards body. The International Biometric Industry Association (IBIA) has agreed to be the organization which will manage the registration and issuance of and archiving of the OBJECT IDENTIFIERs and RELATIVE OBJECT IDENTIFIERs for Organizations and Vendors which require them.

D.2.1 Biometric Processing or Matching Parameters

As stated above, biometric processes only needs to be registered if there are associated parameters that need to be sent. The individual processing parameters do not have to be registered as long as they are defined and maintained by the organization which registered the process. The processing parameters are associated with that particular OBJECT IDENTIFIER.

The application would be responsible for determining compatible versions. If the versions are incompatible, then the processing information may have to be rejected, and therefore the authentication process would have to fail.

Such parameters should and could be standardized upon to reduce the overhead for systems which want to incorporate multiple biometric devices. This is likely to happen in the future as biometric technology matures.

Appendix E AN Example of embedding a CBEFF object:

E. The X.509 AuthenticationInfo Attribute certificate

E.1 Certificate Background

This section has been appended as an example of how the CBEFF can be used to place Biometric Data within an Attribute certificate. It is widely believed that many systems will, in the future, use X.509 certificates to hold biometric templates, therefore this may be an appropriate example.

E.1.1 Attribute Certificates

Attribute certificates are used to convey a set of attributes along with a public key certificate identifier (i.e. a serial number and a public key certificate issuer name) or entity name. The attributes are placed in a separate structure to maintain conformance with existing international standards (X.509). An entity may have multiple attribute certificates associated with each of its public keys certificates.

X9.57, originated by the American Bankers Association (ABA) and adopted by ANSI, also defines an attribute certificate which is complimentary to the X.509 certificate. ISO/IWD-15782-2 also describes the attribute certificate, with added details for banking applications.

There is no requirement that the same authority create both the public key certificate and the attribute certificate; in fact, role separation should frequently dictates otherwise. The generation of an attribute certificate may be requested by an entity other than the subject of the attribute certificate. The X9.57 specification does not define the messages between an entity and the attribute authority (AA) dealing with the generation of the attribute certificate.

X9.57 defines an attribute as information, excluding the public key, which is provided by an entity or an AA and certified by the AA in an attribute certificate. Attributes are bound to a public key certificate or entity name by the signature of the AA on the attribute certificate.

The information contained in the attribute certificate is as follows:

Table B.2 - X9.57 Attribute Certificate Fields

Field	Description
Version	This identifies the version of the attribute certificate.
serial Number	This field uniquely identifies this certificate among all those issued by the AA. (if the AA is also a CA, the serial number space is thus shared by the public key certificates and the attribute certificates.)
owner	An attribute certificate may be linked to either a particular entity, or one of that entity's public key certificates. The mechanism to be used is specified by the application or standard which uses the attribute certificate.
IssuerName	This field contains the name of the issuer of the attribute certificate (an AA).
Issuer Unique Identifier	This field uniquely identifies the issuer, in the case where the issuer name is not sufficient.

Validity	This specifies when a certificate is valid. The period is described by a start date and time and an end date and time as follows: notBefore: The start time that the certificate is valid. notAfter: The end time that the certificate is valid.
Attributes	The attributes are information concerning the entity, or the certification process. They may be supplied by either the entity, a third party entity or the AA depending upon the application.
extension(s)	The extensions field allows addition of new fields to the attribute certificate without modification of the ASN.1 definition.
SignatureAlgorithm	This field identifies the algorithm used to sign the certificate.
Signature	The signature field consists of: The output of the signing function (i.e. the signed hash value of the data in this certificate). This data is used to verify the data in the certificate.

The AttributeCertificate matching rule was created to allow more complex matching than the certificateExactMatch (a matching rule defined in X.509). It allows comparison to the issuer's serialNumber, the owner, the issuerName, and the validity. Refer to X.509 for further information on the matching rules.

E.1.1.1 Attribute Certificate Advantages/Disadvantages

Attribute certificates are essentially X.509 certificates without public key information (alternatively one can perceive them as extended certificates without the X.509 certificate embedded into them.) They are intended to compliment the X.509 certificate with additional information about the user (subject). This would give the same advantages and disadvantages as the PKCS#6 certificate with the additional benefits and disadvantages listed below:

Advantages:

- Mutual verification, via a challenge response, can be performed between the holder of the attribute certificate and the user authenticator prior to sending the attribute information.
- The attribute information can be encrypted, providing access to the confidential information to verified authenticators only.
- Information can be separated into as many attribute certificates as needed by the system. This may be useful in meeting the "need to know" requirement of many systems.
- Anonymity can be accommodated if the Distinguished Name (DN) of the user's X.509 certificate is a reference, not an actual identity (i.e. a user number, database lookup, etc.). The DN can be used to match attribute certificates with X.509 certificates.
- Attribute certificates are becoming standardized (as with X.509).

Disadvantages:

- Introducing multiple attribute authorities into the system architecture makes the system more complex. Key management issues may be prevalent.
- User authentication processing time may be an issue if two signatures must be verified, and the attribute certificate needs to be decrypted.

E.1.2 X.509 Attributes

X.509 imports the attribute definition from X.501.

The X.501 defined attribute (that is AttributeTypeandValue) is as follows:

```
AttributeTypeandValue::=SEQUENCE
   type   Attribute.&id ({SupportedAttributes});
   value ({Attribute.&Type({SupportedAttributes}){@type}}}
```

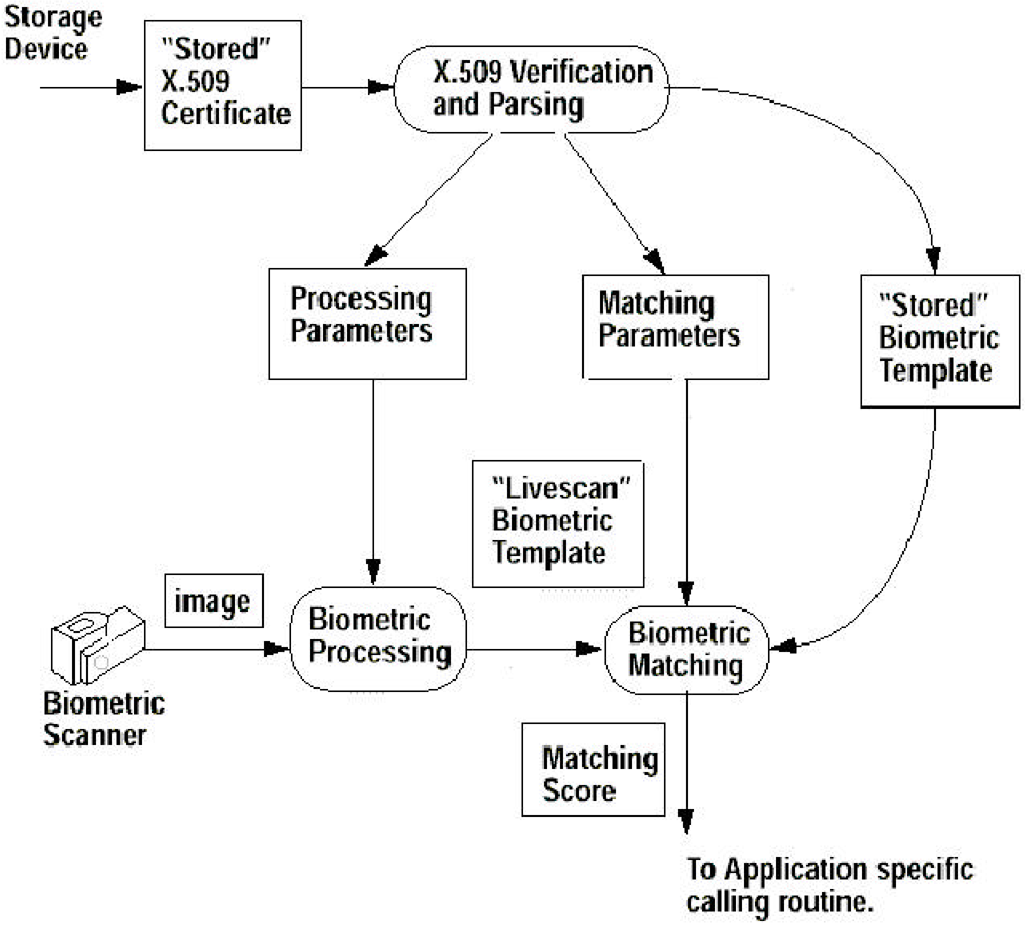
All attributes are assigned an identifier using an object type of id-at. Any registered attribute, assigned a unique identifier by an ISO recognized standards body, can be used. X.520 is a source for ISO defined attributes; however, many other standards body have registered attributes which may used.

The CBEFF Object (the SBH) , as defined in this specification, can be used as an Attribute.The Biometric information can be placed in the CBEFF Object. The CBEFF Object can then be placed within the Attribute Certificate as detailed in the following sections. The X984 BSMB defintion will be used.

E.2 An Example based upon the X9.84 BSMB definition.

The following diagram illustrates how the biometric processing and matching parameters would be utilized during a biometric verification process:

Figure D.1: Using an X.509 Certificate with Detailed Biometric Information


Storage Device
"Stored" X.509 Certificate
X.509 Verification and Parsing
Processing Parameters
Matching Parameters
"Stored" Biometric Template
"Livescan" Biometric Template
image
Biometric Processing
Biometric Matching
Biometric Scanner
Matching Score
To Application specific calling routine.

E.2.1 User Verification

The certificate used to store the biometric information would be transferred to the entity performing the verification (from a database, smartcard, disk, etc.). The entity would verify the signature on the X.509 certificate to detect alteration and to prove the validity of the biometric template. The CBEFF object (The SBH) is a certifcate and can be extracted from the certificate. The SBH is DER decoded using a commercial encode/decode engine. The Biomtric template and each of the processing/matching paremeters can be extracted from the data returned from the engine.

The processing parameters are fed to the biometric processing function which converts the livescan image to a livescan biometric template. The livescan biometric template, the biometric template from the X.509 certificate and the matching algorithm parameters from the X.509 certificate are fed into the matching algorithm for verification of the user. The result of that operation should indicate the authenticity of the claimed identity of the user.

E.3 ASN.1 Authentication Attribute Certificate definition

The attribute certificate that holds the authentication information attribute is described in ASN.1 as follows [ISO/IEC 9594-8:1997]:

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE {
            version   Version DEFAULT v1,
            subject   CHOICE {
            baseCertificateID[0]      IssuerSerial, -- associated with a Public Key Certificate
                     subjectName[1]GeneralNames },  -- associated with a name
            issuer                     GeneralNames, -- CA issuing the attribute certificate
            signature                  AlgorithmIdentifier,
            serialNumber              CertificateSerialNumber,
            attrCertValidityPeriod    AttCertValidityPeriod,
            StandardBiometic          SBH,
            UniqueissuerID,
            issuerUniqueID            UniqueIdentifier OPTIONAL,
            extensions                 Extensions OPTIONAL}

IssuerSerial  ::= SEQUENCE {
            issuer          GeneralNames,
            serial                    CertificateSerialNumber,
            issuerUID                 UniqueIdentifier OPTIONAL}
```

```
AttCertValidityPeriod ::= SEQUENCE {

            notBeforeTime   GeneralizedTime,
            notAfterTime    Generalized Time }
```

E.4 Approximate Certificate Data Size

An approximation of data sizes can be made on the following assumptions.

- The size of the signature and public key info is set at 512 bits (64 octets where 1octet = 1 byte).
- No extensions are used.
- Distinguished Encoding Rules (DER) is utilized by the CA signature certificates and user certificate.

Table B.6 - Contents of Attribute Cert - Identification & Authentication Cert

Item	item size	# of items	Total Size
Version	**5 octets**	**1**	**5 octets**
Owner (baseCertificateID)	**8 octets**	**1**	**8 octets**
issuer (AA)	**183 octets**	**1**	**183 octets**
signature	**9 octets**	**1**	**9 octets**
serialNumber	**6 octets**	**1**	**6 octets**
validity	**32 octets**	**1**	**32 octets**
AuthenticationInfo -biometricInfo	**500 octets**	**1**	**500 octets**
IssuerUniqueID (Token Serial #)	**16 octets**	**1**	**16 octets**
AlgorithmIdentifier	**9 octets**	**1**	**9 octets**
signatureValue	**70 octets**	**1**	**70 octets**
Total			**838 octets**

If additional fields are added (such as extensions) simply add the length of the new field to the total.